

Membangun Website Pribadi dengan Next.js (App Router), React, dan Tailwind CSS

Tujuan Pembelajaran

1. Memahami dan mengimplementasikan **App Router** di Next.js.
2. Membuat komponen React yang reusable.
3. Menerapkan **Server-Side Rendering (SSR)** dan **Static Site Generation (SSG)**.
4. Menggunakan Tailwind CSS untuk styling dan desain responsif.

Alat dan Bahan

1. Komputer dengan sistem operasi Windows, macOS, atau Linux.
2. Node.js dan npm (Node Package Manager) terinstal.
3. Text editor seperti Visual Studio Code.
4. Browser modern (Chrome, Firefox, Edge, dll.).

Langkah-langkah Praktikum

1. Persiapan Lingkungan

1. Pastikan Node.js dan npm sudah terinstal di komputer Anda. Anda dapat memeriksanya dengan menjalankan perintah berikut di terminal atau command prompt:

```
node -v
```

```
npm -v
```

2. Buat direktori baru untuk proyek Next.js Anda:

```
mkdir website-pribadi
```

```
cd website-pribadi
```

3. Inisialisasi proyek Next.js dengan TypeScript dan App Router:

```
npx create-next-app@latest . --typescript --app
```

```

C:\Users\Acer\projects pbf\website-pribadi>npx create-next-app@latest . --typescript
Need to install the following packages:
create-next-app@15.2.1
Ok to proceed? (y) y

✓ Would you like to use ESLint? ... No / Yes
✓ Would you like to use Tailwind CSS? ... No / Yes
✓ Would you like your code inside a `src/` directory? ... No / Yes
✓ Would you like to use App Router? (recommended) ... No / Yes
✓ Would you like to use Turbopack for `next dev`? ... No / Yes
✓ Would you like to customize the import alias (`@/*` by default)? ... No / Yes
Creating a new Next.js app in C:\Users\Acer\projects pbf\website-pribadi.

Using npm.

Initializing project with template: app-tw

Installing dependencies...

```

4. Cek konfigurasi postcss.config.mjs:

```

JS postcss.config.mjs > ...
1  const config = {
2    |   plugins: ["@tailwindcss/postcss"],
3  };
4
5  export default config;
6

```

5. Cek di app/globals.css, sudah ada import untuk tailwindcss:

```

src > app > # globals.css > ...
1  @import "tailwindcss";
2
3
4  :root {
5    |   --background: ☐ #ffffff;
6    |   --foreground: ☒ #171717;
7  }
8
9  @theme inline {
10 |   --color-background: var(--background);

```

6. Buka app/layout.tsx, tambahkan import './globals.css', dan modifikasi menjadi sebagai berikut:

```

src > app > layout.tsx > ...
1  import Layout from '../components/Layout';
2  import './globals.css'
3
4  export default function RootLayout({
5    children,
6  }): {
7    children: React.ReactNode
8  } {
9    return (
10     <html lang="en">
11       <body>
12         <Layout pageId="about_me">
13           {children}
14         </Layout>
15       </body>
16     </html>
17   );
18 }

```

7. Jalankan aplikasi Next.js:

npm run dev

Aplikasi akan terbuka di browser pada alamat <http://localhost:3000>.

2. Membuat Halaman Website

Website pribadi akan memiliki tiga halaman:

- **Tentang Saya:** Halaman berisi profil singkat dan link ke media sosial.
- **Proyek:** Halaman untuk menampilkan proyek-proyek yang telah diselesaikan.
- **Esai:** Halaman untuk menampilkan daftar artikel yang telah ditulis.

Langkah-langkah:

1. Buat file `app/page.tsx` sebagai halaman "Tentang Saya":

```
src > app > page.tsx > Home
1 export default function Home() {
2   |   return <div>Tentang Saya</div>;
3   }
```

2. Buat file `app/projects/page.tsx` sebagai halaman "Proyek":

```
src > app > projects > page.tsx > Projects
1 export default function Projects() {
2   |   return <div>Proyek</div>;
3   }
```

3. Buat file `app/essays/page.tsx` sebagai halaman "Esai":

```
src > app > essays > page.tsx > Essays
1 export default function Essays() {
2   |   return <div>Esai</div>;
3   }
```

4. Buka browser dan akses:
 - o `http://localhost:3000/` untuk halaman "Tentang Saya".
 - o `http://localhost:3000/projects` untuk halaman "Proyek".
 - o `http://localhost:3000/essays` untuk halaman "Esai".

3. Membuat Layout dan Navigasi

Layout digunakan untuk menyediakan struktur yang konsisten di seluruh halaman, seperti navbar dan footer.

Langkah-langkah:

1. Buat direktori `src/components` jika belum ada.
2. Buat file `src/components/Layout.tsx`:

```

src > components > Layout.tsx > Layout
1  import Head from 'next/head';
2  import Navbar from './Navbar';
3  import Footer from './Footer';
4
5  type LayoutProps = {
6    |   children: React.ReactNode;
7  };
8
9  export default function Layout({ children }: LayoutProps) {
10    return (
11      <>
12        <Head>
13          <link rel="icon" href="/favicon.ico" />
14        </Head>
15        <Navbar />
16        <main>{children}</main>
17        <Footer />
18      </>
19    );
20  }

```

3. Buat file `src/components/Navbar.tsx`

```

'use client';

import { usePathname } from 'next/navigation';

type NavItemProps = {
  title: string;
  url: string;
  isSelected: boolean;
};

function NavItem({ title, url, isSelected }: NavItemProps) {
  return (
    <li>
      <a
        className={`block px-3 py-2 transition hover:text-teal-500 ${
          isSelected ? "text-teal-500" : ""
        }`}
        href={url}
      >
        {title}
      </a>
    </li>
  );
}

export default function Navbar() {
  const pathname = usePathname();

  return (
    <div className="flex justify-center mx-auto max-w-7xl h-16 pt-6">
      <nav>

```

```

        <ul className="flex rounded-full bg-white/90 px-3 text-sm font-medium text-zinc-800 shadow-lg shadow-zinc-800/5 ring-1 ring-zinc-900/5 backdrop-blur">
          <NavItem
            title="Tentang Saya"
            url="/"
            isSelected={pathname === '/'}
          />
          <NavItem
            title="Proyek"
            url="/projects"
            isSelected={pathname === '/projects'}
          />
          <NavItem
            title="Esai"
            url="/essays"
            isSelected={pathname === '/essays'}
          />
        </ul>
      </nav>
    </div>
  );
}

```

4. Buat file `src/components/Footer.tsx`

```

src > components > Footer.tsx > ...
1
2  type FooterLinkProps = {
3    text: string;
4    url: string;
5  };
6
7  function FooterLink({ text, url }: FooterLinkProps) {
8    return <a className="transition hover:text-teal-500" href={url}>{text}</a>;
9  }
10
11  export default function Footer() {
12    return (
13      <footer className="pt-10 px-8 pb-16 border-t">
14        <div className="flex justify-between gap-6">
15          <div className="flex gap-6 text-sm font-medium text-zinc-600">
16            <FooterLink text="Tentang Saya" url="/" />
17            <FooterLink text="Proyek" url="/projects" />
18            <FooterLink text="Esai" url="/essays" />
19          </div>
20          <p className="text-sm text-zinc-400">© 2023 Nama Anda. All rights reserved.</p>
21        </div>
22      </footer>
23    );
24  }
25

```

5. Update file `app/layout.tsx` untuk menggunakan layout:

```

src > app > layout.tsx > RootLayout
1  import type { Metadata } from 'next';
2  import { Inter } from 'next/font/google';
3  import './globals.css';
4  import Layout from '../components/Layout';
5
6  const inter = Inter({ subsets: ['latin'] });
7
8  export const metadata: Metadata = {
9    title: 'Website Pribadi',
10   description: 'Website pribadi untuk menampilkan proyek dan esai.',
11 };
12
13 export default function RootLayout({
14   children,
15 }: {
16   children: React.ReactNode;
17 }) {
18   return (
19     <html lang="en">
20       <body className={inter.className}>
21         <Layout>
22           {children}
23         </Layout>
24       </body>
25     </html>
26   );
27 }

```

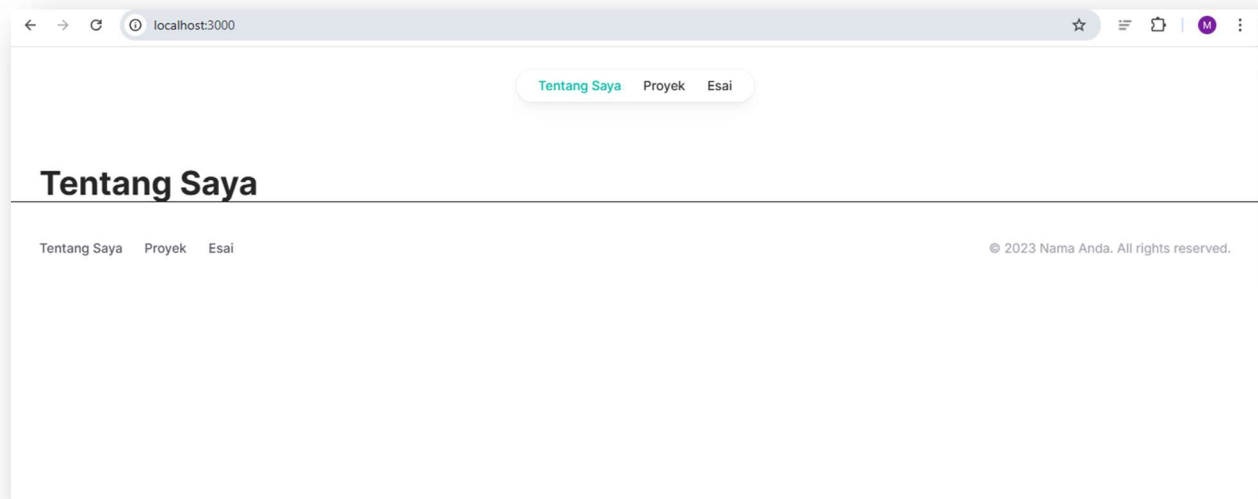
6. Update setiap halaman menambahkan metadata dengan generateMetadata:

```

src > app > page.tsx > metadata > openGraph
1  import type { Metadata } from 'next';
2
3  export const metadata: Metadata = {
4    title: 'Tentang Saya',
5    description: 'Halaman tentang saya.',
6    openGraph: {
7      title: 'Tentang Saya',
8      description: 'Halaman tentang saya.',
9    },
10 };
11
12 export default function Home() {
13   return (
14     <div className="mt-16 px-8">
15       <header>
16         <h1 className="font-bold text-4xl text-zinc-800">Tentang Saya</h1>
17       </header>
18     </div>
19   );
20 }

```

Lakukan hal yang sama untuk halaman **projects/page.tsx** dan **essays/page.tsx**.



4. Membuat Halaman Proyek dengan Grid Responsif

Halaman proyek akan menampilkan daftar proyek dalam bentuk grid yang responsif.

Langkah-langkah:

1. Buat folder di **public/images**. Kemudian tambahkan dua image, lalu rename dengan nama **project1.png** dan **project2.png**

2. Modifikasi file `app/projects/page.tsx`:

Tambahkan import `next/image` dan beberapa `image` yang telah ditambahkan sebelumnya

```
2 import Image, { StaticImageData } from 'next/image';
3 import image1 from '../../../public/images/project1.png';
4 import image2 from '../../../public/images/project2.png';
5
```

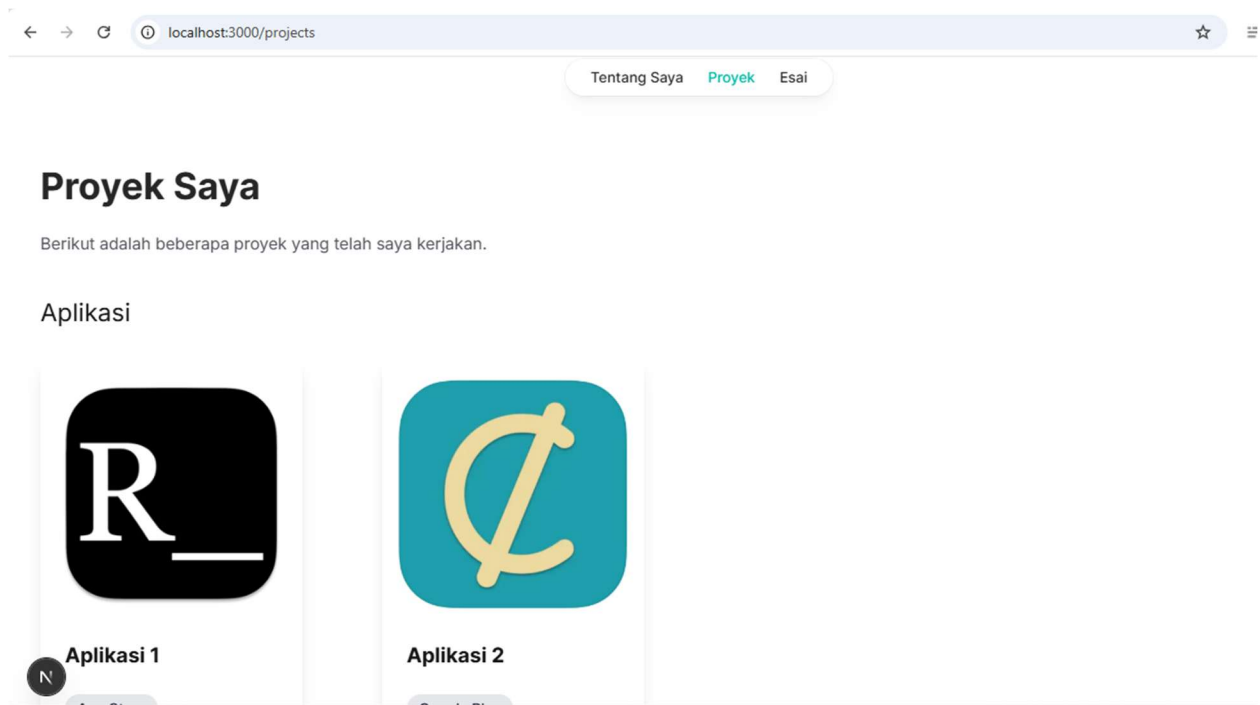
Buat `card project item` sebagai berikut:

```
16 type ProjectItemProps = {
17   name: string;
18   url: string;
19   urlDisplay: string;
20   imageSrc: StaticImageData;
21 };
22
23 function ProjectItem({ name, url, urlDisplay, imageSrc }: ProjectItemProps) {
24   return (
25     <li>
26       <a href={url}>
27         <div className="max-w-sm rounded overflow-hidden shadow-lg">
28           <Image className="w-full" src={imageSrc} alt={name} />
29           <div className="px-6 py-4">
30             <div className="font-bold text-xl mb-2">{name}</div>
31           </div>
32           <div className="px-6 pb-4">
33             <span className="inline-block bg-gray-200 rounded-full px-3 py-1 text-sm font-semibold text-gray-700 mr-2 mb-2">
34               {urlDisplay}</span>
35           </div>
36         </div>
37       </a>
38     </li>
39   );
40 }
```

Modifikasi komponen `project` sebagai berikut:

```
42 export default function Projects() {
43   return (
44     <div className="mt-16 px-8">
45       <header>
46         <h1 className="font-bold text-4xl text-zinc-800">Proyek Saya</h1>
47         <p className="text-base mt-6 text-zinc-600">Berikut adalah beberapa proyek yang telah saya kerjakan.</p>
48       </header>
49       <div className="mt-16">
50         <h2 className="text-2xl">Aplikasi</h2>
51         <ul className="grid grid-cols-1 md:grid-cols-2 lg:grid-cols-4 gap-x-12 gap-y-16 mt-8">
52           <ProjectItem name="Aplikasi 1" url="https://example.com" urlDisplay="App Store" imageSrc={image1} />
53           <ProjectItem name="Aplikasi 2" url="https://example.com" urlDisplay="Google Play" imageSrc={image2} />
54         </ul>
55       </div>
56     </div>
57   );
58 }
```

3. Simpan file dan buka `http://localhost:3000/projects` di browser. Anda akan melihat daftar proyek dalam grid yang responsif.



Tugas

Eksplorasi Tailwindcss, kemudian.

1. Modifikasi halaman "Esai" dengan Grid daftar artikel yang telah ditulis.
2. Modifikasi halaman "Tentang Saya" Buat tampilan yang menarik

Kesimpulan

Dalam praktikum ini, Anda telah mempelajari cara membangun website pribadi menggunakan Next.js (App Router), React, dan Tailwind CSS. Anda juga telah mengimplementasikan fitur-fitur seperti routing, layout, serta komponen reusable.

Referensi

- Dokumentasi Resmi Next.js: <https://nextjs.org/docs>
- Dokumentasi Tailwind CSS: <https://tailwindcss.com/docs>